TYPOO

A PROJECT REPORT

submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
CSE with Specialization in
Cloud Computing & Virtualization

Submitted by:

Name	Course	Branch	SAP ID
Prabal Bansal	B.Tech. CSE	CCVT	500068861
Raghav Jindal	B.Tech. CSE	CCVT	500067476
Raman Kumar	B.Tech. CSE	CCVT	500078477
Sachin Agrawal	B.Tech. CSE	CCVT	500067770

Under the guidance of

Mr. Abhirup Khanna
Assistant Professor, Department of Virtualization

SCHOOL OF COMPUTER SCIENCE
UNIVERSITY OF PETROLEUM & ENERGY STUDIES
Bidholi Campus, Energy Acres, Dehradun – 248007

2021-22

CANDIDATES' DECLARATION

I/We hereby certify that the project work entitled **TYPOO** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology inComputer Science And Engineering with Specialization in Cloud Computing and VirtualizationTechnology, and submitted to the Department of Virtualization at School of Computer Science,University of Petroleum And Energy Studies, Dehradun, is an authentic record of my/our work carried out during the period from **January, 2022** to **May, 2022** under the supervision **of Mr. Abhirup Khanna**, Assistant Professor, Department of Virtualization.

The matter presented in this project has not been submitted by me/us for the award of any other degree of this or any other University.

Prabal Bansal (R110218104)
Raghav Jindal (R110218114)
Raman Kumar (R110218119)
Sachin Agrawal (R110218129)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Date: 5th May, 2022)

Mr. Abhirup Khanna
(Project Guide)

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide **Mr. Abhirup Khanna**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

Prabal Bansal (R110218104)
Raghav Jindal (R110218114)
Raman Kumar (R110218119)
Sachin Agrawal (R110218129)

School of Computer Science

University of Petroleum & Energy Studies, Dehradun

Major: II

PROJECT TITLE: TYPOO

ABSTRACT

A full-stack blogging web application made with node and express js and backend with MongoDB with all the CRUD operations. Users can register themselves and write their own blog from their dashboard and the admin will be able to feature that blog on the running website. We also added the functionality of listing blog according to certain topic or interest in the home page itself which will add more look and field to the application.
Users can register them in the application with the social media as well as mobile number. For social media we have added the functionality of sign in with google and Facebook. And for the mobile number functionality we have added the twillio which will send an OTP the registered mobile number.

Users can update their profile data with all the basic details like the name and password, Users can also update their profile photo by selecting a png file from their system.

Keywords: Full-Stack , Database, React, Front-end .

TABLE OF CONTENT

TABLE OF FIGURES

LIST OF TABLES

INTRODUCTION

As we know that most of the platforms are build up of combination of multiple component relying on different middleware that are API's.

Due to these API's a lot of work get easier as a lot of complex task are handled by these middleware.
People are able to develop product in short period of time with the help of them and are able to deliver the product more quickly as compared to when they have to create the product from scratch comprising all the features as well as all the different component of a product.

Keeping that in mind and further integrating with components like Node.js which is capable of server side scripting and MongoDB capable of fetching multiple request at the same time and giving a lot of flexibility with the Database to play around , combined these we can create applications that are more reliable and useful at the same time and can be scaled at any moment of time.

A full-stack blogging web application made with node and express js and backend with MongoDB with all the CRUD operations. Users can register themselves and write their own blog from their dashboard and the admin will be able to feature that blog on the running website. We also added the functionality of listing blog according to certain topic or interest in the home page itself which will add more look and field to the application.

Blogging has become such a mania that a new blog is being created every second of every minute of every hour of every day. A blog is your best bet for a voice among the online crowd. Blogs were usually the work of a single individual occasionally of a small group, and often covered a single subject. More recently, "multi-author blogs" (MABs) have developed, with posts written by large numbers of authors and professionally edited. MABs from newspapers, other media outlets, universities, think tanks, advocacy groups, and similar institutions account for an increasing quantity of blog traffic.

WHY WE CHOOSE THIS PROJECT:-

In recent past time Blogs are store in the paper files and difficult to search or modify any information, for expanding the Blogs infrastructure, Awareness of environmental issues or any other issues related to education, health, digital technology, and search for greater safety give to information to all persons in all age groups and a new role within the education system, I choose this project. As a result of these project initiatives phenomenal growth has taken place in all the activities of blogs and any user can share its information related to any topic to all users.

Benefits of Project:-

This is a very simple design and implement. It has got following features:

- Data can be saved safely.
- No other person cannot view other person's detail
- Greater efficiency
- User friendliness
- Free of Cost
- Lets User know others opinion through comments.

Applications:-

TYPOO enables the users to create innovative and attractive information with the text and categories in just few simple steps. The user just needs to upload some images of his choice and can also upload the information or can select from the given category list. This website will provide a personalized environment that would contain the data in motion with images.

Scope of the Project:-

- To Share Technical knowledge
- Contributing to open source
- Share your opinion on internet
- Posting comments.
- To merge users according to certain interest
- Help people to collaborate.

SYSTEM MODULE:-

The modules involved in this project are:

- User
- Admin

User In this module

- User can signup
- User can login
- User can upload multiple images to his profile
- User can add information/Blog on certain categories
- User can add comments
- User can select any categories

Admin In this module

- Admin can block user id
- Admin can see all pages
- Admin can create more Categories
- Admin can maintain all records of user

PROBLEM STATEMENT

- With this project we are trying to bring together communities with intellect in different fields having same interest of knowledge.
- This project is open to people who wants to share their knowledge in a particular field.
- To share individual thoughts on the internet.

LITERATURE REVIEW

[1] In this paper, Twillio is used for authentication with SMS. Passwords get stolen every day. In 2018, hackers swiped 2.5 billion accounts—that amounts to around 6.85 million stolen passwords per day and 158 per second. While that number might sound alarming (because it is), it's significantly less than 2.5 billion. With SMS verification enabled, a hacker would need your username, password, *and* access to your phone (and they might even need a password to unlock your phone). After entering your username and password, companies will send an SMS verification code to your smartphone. Use that code to complete your login—this process is SMS verification.

[2] In this paper, Facebook authentication is used for registering users. The first time you log into the React app with Facebook, an account is registered that is associated with your Facebook Id so the app can identify you when you login again with Facebook. The account is created with the name from your Facebook account and an extra Info field with some default text, both the name and extra info can be updated in the React app, and updating account details only changes them in the app it doesn't affect anything on Facebook.

[3] React is used for the front end, Both HTML and CSS are integral to any web development project. If you have these skills already, then learning React should be a relatively straightforward process. It has its own unique set of challenges, but it is an excellent tool to have in order to start or further your career as a web developer. We rely on the power of JavaScript to generate HTML that depends on some data, rather than enhancing HTML to make it work with that data. Enhancing HTML is what other JavaScript frameworks usually do.

[1] This paper Node JS is used for handling the backed. Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node. js is primarily used for non-blocking, event-driven servers, due to its single-threaded nature. It's used for traditional web sites and back-end API services, but was designed with real-time, push-based architectures in mind. java uses the concept of multithreading with ease, whereas Node JS does not use the concept of multi-threading like Java does. For large scale projects that involved concurrency, Java is highly recommended, whereas Node JS does not handle the thread and Java, which is the weakest point of this framework.

[2] In this paper, npm is used for connection with database, npm is a package manager for the JavaScript programming language maintained by npm, Inc. npm is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry. npm is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently. It is extremely configurable to support a wide variety of use cases. Most commonly, it is used **to publish, discover, install, and develop node programs**.

[1] For the database MongoDb is used, MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License. **MongoDB offers faster query processing but with an increased load and system requirements**. Without knowing the purpose of use, it is not possible to classify SQL Databases or NoSQL Databases like MongoDB as better or worse than the other. There are various factors that drive the MongoDB vs SQL decision MongoDB is built on a scale-out architecture that has become popular with developers of all kinds for **developing scalable applications with evolving data schemas**. As a document database, MongoDB makes it easy for developers to store structured or unstructured data. It uses a JSON-like format to store documents. NoSQL databases come in a variety of types including document databases, key-values databases, wide-column stores, and graph databases. MongoDB is the world's most popular **NoSQL database. Network encryption is available with MongoDB**. This allows you to protect your database and communications through an industry-standard encryption methodology. TLS and SSL are supported by the x. 509 certificates, which clients can use to authenticate their identities. MongoDB is a document-based non-relational database management system. It's also called an object-based system. It was designed to supplant the MySQL structure as an easier way to work with data. On the other hand, MySQL is a table-based system (or open-source relational database)

[1] APIs are used for fetching data, An application programming interface is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build or use such a connection or interface is called an API specification. API stands for "application programming interface." An API is essentially **a set of rules that dictate how two machines talk to each other**. Some examples of API-based interactions include a cloud application communicating with a server, servers pinging each other, or applications interacting with an operating system. **The easiest way to develop your API is to use a tool**. For instance, you can build your API using Akana. With Akana, you have two ways to develop your API: Create an API that exposes an existing resource. APIs are needed **to bring applications together in order to perform a designed function built around sharing data and executing pre-defined processes**. They work as the middle man, allowing developers to build new programmatic interactions between the various applications people and businesses use on a daily basis.

[2] POSTMAN is used for testing the APIs, Postman is an API platform for developers to design, build, test and iterate their APIs. As of April 2022, Postman reports having more than 20 million registered users and 75,000 open APIs, which it says constitutes the world's largest public API hub. Postman is an application used for **API testing**. It is an HTTP client that tests HTTP requests, utilizing a graphical user interface, through which we obtain different types of responses that need to be subsequently validated. Postman is **an API client that makes it easy for developers to create, share, test and document**

[1] APIs are used for fetching data, An application programming interface is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build or use such a connection or interface is called an API specification. API stands for "application programming interface." An API is essentially **a set of rules that dictate how two machines talk to each other**. Some examples of API-based interactions include a cloud application communicating with a server, servers pinging each other, or applications interacting with an operating system. **The easiest way to develop your API is to use a tool**. For instance, you can build your API using Akana. With Akana, you have two ways to develop your API: Create an API that exposes an existing resource. APIs are needed **to bring applications together in order to perform a designed function built around sharing data and executing pre-defined processes**. They work as the middle man, allowing developers to build new programmatic interactions between the various applications people and businesses use on a daily basis.

[2] POSTMAN is used for testing the APIs, Postman is an API platform for developers to design, build, test and iterate their APIs. As of April 2022, Postman reports having more than 20 million registered users and 75,000 open APIs, which it says constitutes the world's largest public API hub. Postman is an application used for **API testing**. It is an HTTP client that tests HTTP requests, utilizing a graphical user interface, through which we obtain different types of responses that need to be subsequently validated. Postman is **an API client that makes it easy for developers to create, share, test and document**

[1] **APIs**. This is done by allowing users to create and save simple and complex HTTP/s requests, as well as read their responses. The result - more efficient and less tedious work. REST is an acronym for Representational State Transfer. It is an architectural style for distributed hypermedia systems. An API is an application programming interface. It is a set of rules that allow programs to talk to each other. The developer creates the API on the server and allows the client to talk to it.

[2] Vs Code is used for code development, Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Visual Studio Code is **a streamlined code editor with support for development operations like debugging, task running, and version control**. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE. In contrast, **Visual Studio Code can be classed as an integrated development environment (IDE)**, meaning that developers can write and test code at the same time.

[3] Express is used as a backend framework, Express.js, or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js. The primary use of Express is **to provide server-side logic for web and mobile**

[1] **applications**, and as such it's used all over the place. Companies which use Express as a foundation of their internet presence include: Accenture.

[2] Body parser is used a syntax handler in backend, body-parser **extracts the entire body portion of an incoming request stream and exposes it on req. body**. The middleware was a part of Express. js earlier but now you have to install it separately. This body-parser module parses the JSON, buffer, string and URL encoded data submitted using HTTP POST request.

[3] Mongoose is used for continuously running of the server, Mongoose is a JavaScript object-oriented programming library that creates a connection between MongoDB and the Express web application framework. Mongoose is **a Node. js-based Object Data Modeling (ODM) library for MongoDB**. It is akin to an Object Relational Mapper (ORM) such as SQLAlchemy for traditional SQL databases. The problem that Mongoose aims to solve is allowing developers to enforce a specific schema at the application layer.

OBJECTIVES

Now the final object here will be to make a dashboard that will let the people do ---

1) Register, login with Email or Phone number.
2) Quick login with Google, Facebook, SMS.
3) Forgot password, reset password and register a new account by Email or SMS verification.
4) Update personal information (name, password and avatar)
5) Create new blog with React quill.
6) Comment real-time with Socket.io
7) Pagination, search with autocomplete Mongodb

SOFTWARE / HARDWARE REQUIREMENTS

Software Requirements:-

a) **Vs Code**
b) **Npm**
c) **nodeJs**
d) **ReactJs**
e) **MongoDb**
f) **MOngoDb Atlas**
g) **GitHub**
h) **Postman**

Hardware Requirements:-

a) **Minimum Hardware Configurations :-**
Processor: 800MHz Intel Pentium III or equivalent
Memory: 512 MB
Disk space: 750 MB of free disk space
Screen resolution is 1024x768 pixels

b) **Recommended Hardware Configurations**
Processor: Intel Core i5 or equivalent.
Memory: 2 GB (32-bit), 4 GB (64-bit)
Disk space: 1.5 GB of free disk space
Screen resolution is 1024x768 pixels

METHODOLOGY

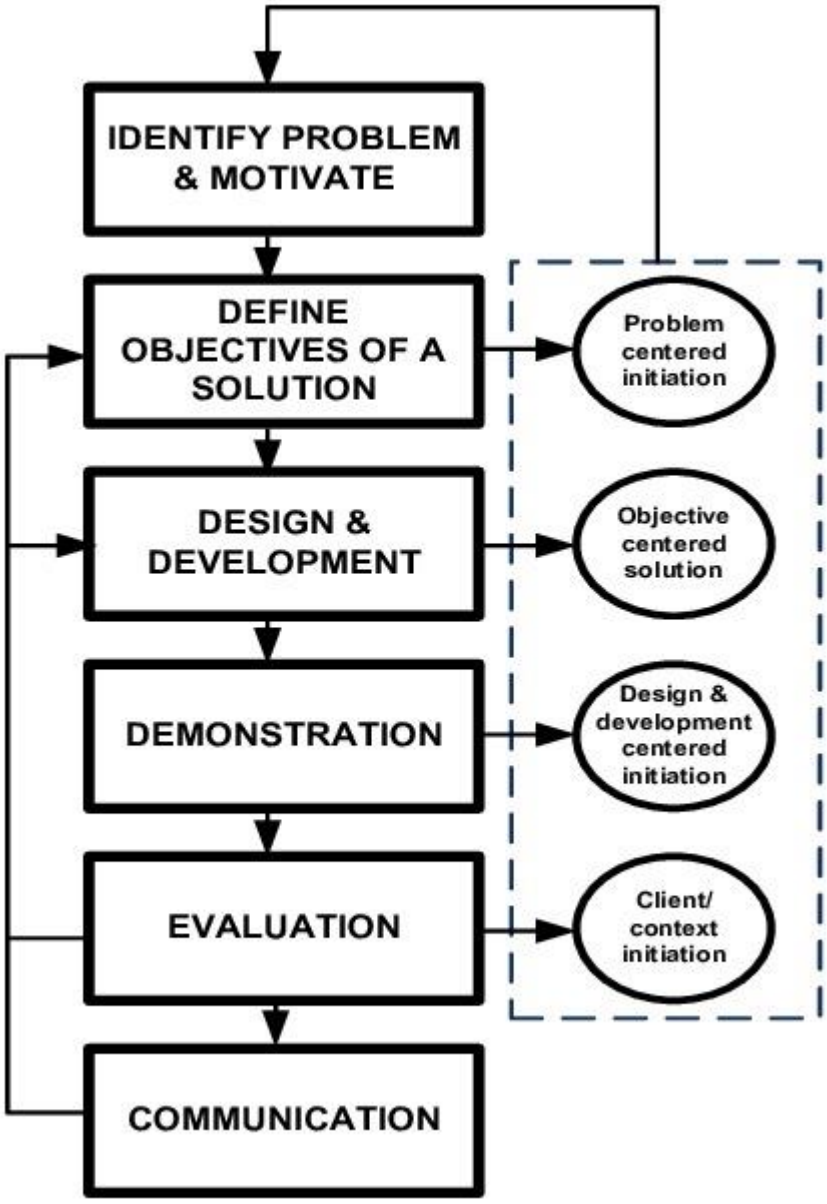


Fig.1 Methodology

The entire implementation of this project can be summarized into the following steps:

1. **Agile methodology** of software development will be followed for the proposed project.

The project is divided into **12 sprints** where the sprint 7 and 8 will consist of parallel development by different members of the team.

2. Each sprint is provided ample time to complete itself as well as to maintain the product's backlog (if any).
3. The project can **accommodate changes** if required at any stage of the project. The sprints 1, 2 and 3 are specifically for requirement analysis and designing of the project. One sprint is specifically designed for setting the environment like maintaining the **Version Control (Git in our case) .**
4. Each development sprint is followed by **Unit Testing** and an **Integration Testing** at the end. **Sprints** are also designed for the reviewing as well as retrospection part.
5. Overall, the time for the project is dedicated to an approach where the beginning time is dedicated towards the requirement analysis and the documentation part and during the implementation part all the team members are following their dedicated sprints cycles to implement the functionality.
6. After the implementation, testing is to be done for the whole application. Finally, the application is deployed with the documentation.

Agile methodology of software development will be followed for the proposed project.

The project is divided into **12 sprints** where the sprint 7 and 8 will consist of parallel development by different members of the team. Each sprint is provided ample time to complete itself as well as to maintain the product's backlog (if any). The project can **accommodate changes** if required at any stage of the project. The sprints 1, 2 and 3 are specifically for requirement analysis and designing of the project. One sprint is specifically designed for setting the environment like maintaining the **Version Control (Git in our case) .**

Each development sprint is followed by **Unit Testing** and an **Integration Testing** at the end. **Sprints** are also designed for the reviewing as well as retrospection part.

Overall, the time for the project is dedicated to an approach where the beginning time is dedicated towards the requirement analysis and the documentation part and during the implementation part all the team members are following their dedicated sprints cycles to implement the functionality. After the implementation, testing is to be done for the whole application. Finally, the application is deployed with the documentation

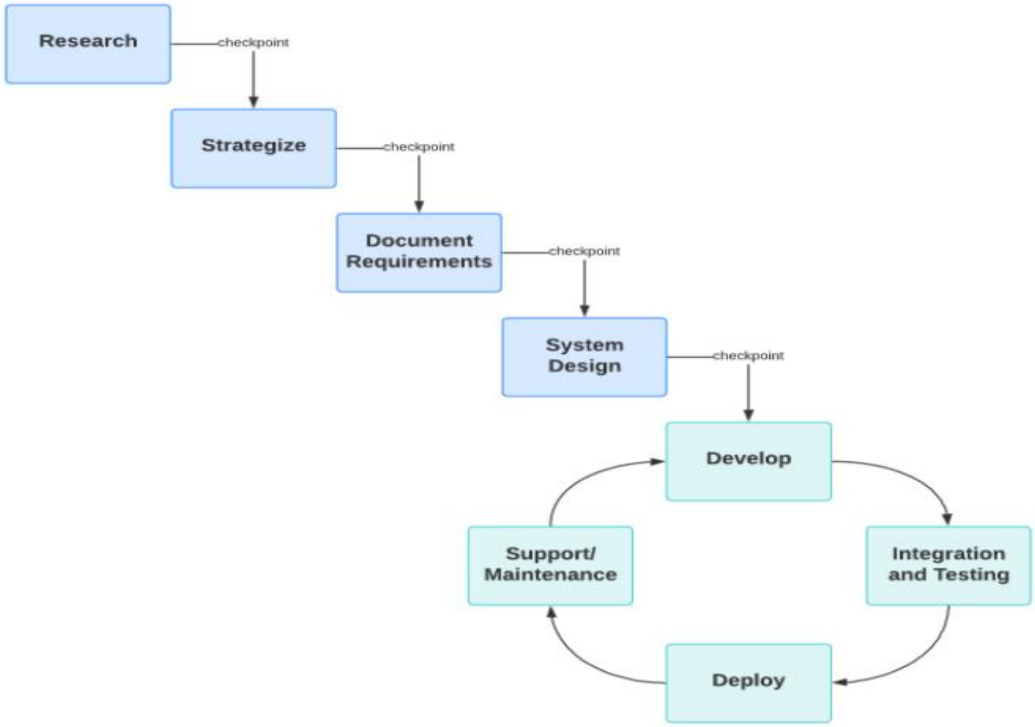


Fig.2 Agile

WORKFLOW

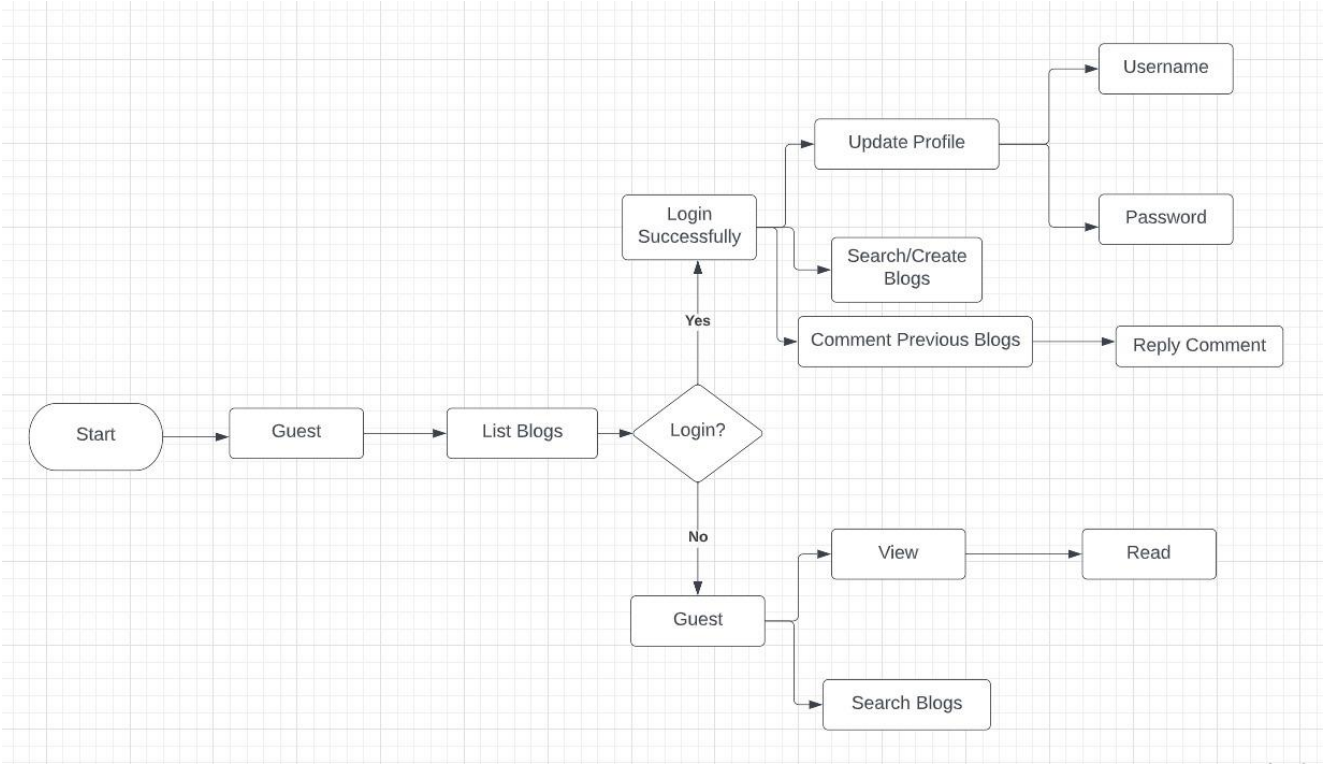


Fig.3 Workflow

For Web Application's User Interface:

1. First, User gets to see the home page where all the featured blogs are present according to the category
2. User can register himself with all the three options available
3. Users data gets stored in mongodb
4. User can then change his/her profile settings
5. User can post blogs according to the different categories
6. User can post blogs on others posts.

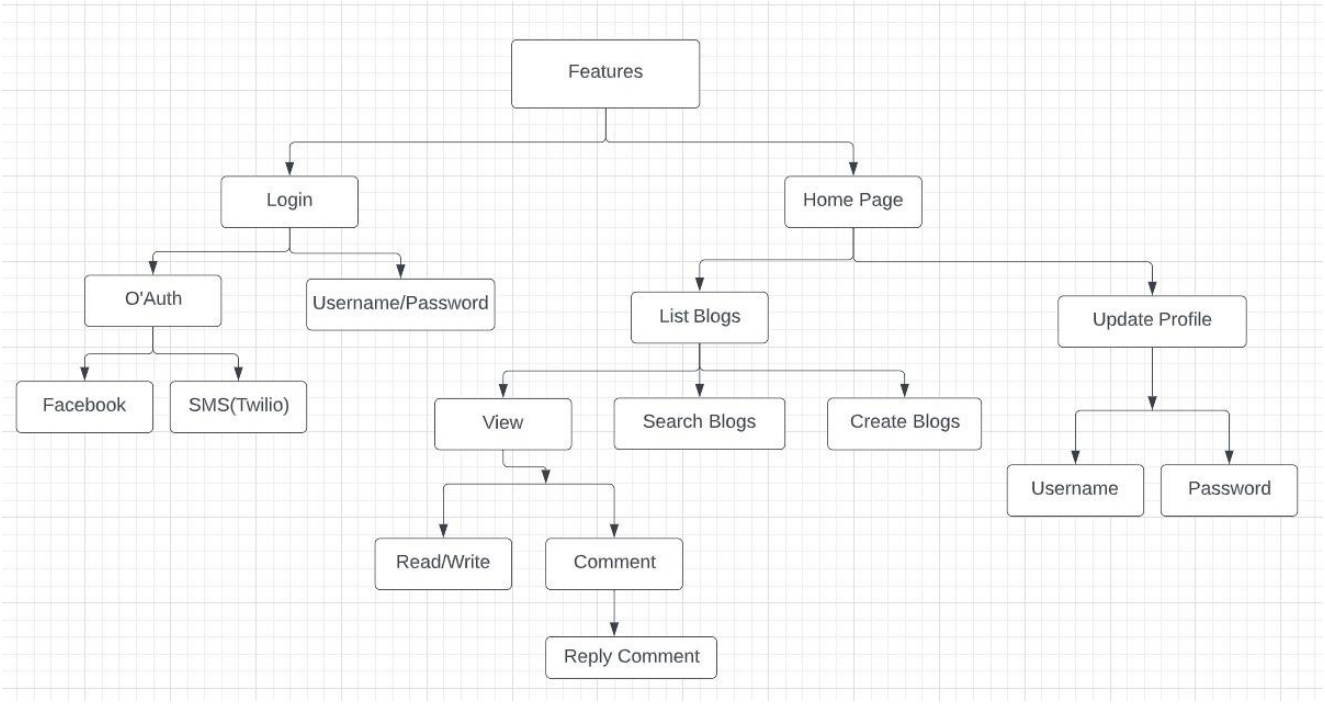


Fig.4 UI Dataflow

Tech Stack Used:

Full Stack

Full stack development: It refers to the development of both **front end**(client side) and **back end**(server side) portions of web application.
Full stack web Developers: Full stack web developers have the ability to design complete web applications and websites. They work on the frontend, backend, database and debugging of web applications or websites.

Front end: It is the visible part of website or web application which is responsible for user experience. The user directly interacts with the front end portion of the web application or website.
Front end Languages: The front end portion is built by using some languages which are discussed below:

- **HTML:** HTML stands for Hyper Text Markup Language. It is used to design the front end portion of web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text documentation within tag which defines the structure of web pages.
- **CSS:** Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.
- **JavaScript:** JavaScript is a famous scripting language used to create the magic on the sites to make the site interactive for the user. It is used to enhancing the functionality of a website to running cool games and web-based software.

Front End Frameworks and Libraries:

- AngularJS**:** AngularJs is a JavaScript open source front-end framework that is mainly used to develop single page web applications(SPAs). It is a continuously growing and expanding framework which provides better ways for developing web applications. It changes the static HTML to dynamic HTML. It is an open source project which can be freely used and changed by anyone.

It extends HTML attributes with Directives, and data is bound with HTML.

- React.js**:** React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front end library responsible only for the view layer of the application. It is maintained by Facebook.
- Bootstrap**:** Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.
- jQuery**:** jQuery is an open source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript. Elaborating the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.

MERN Stack

MERN Stack: MERN Stack is a Javascript Stack that is used for easier and faster deployment of full-stack web applications. MERN Stack comprises of 4 technologies namely: MongoDB, Express, React and Node.js. It is designed to make the development process smoother and easier.
Each of these 4 powerful technologies provides an end-to-end framework for the developers to work in and each of these technologies play a big part in the development of web applications.

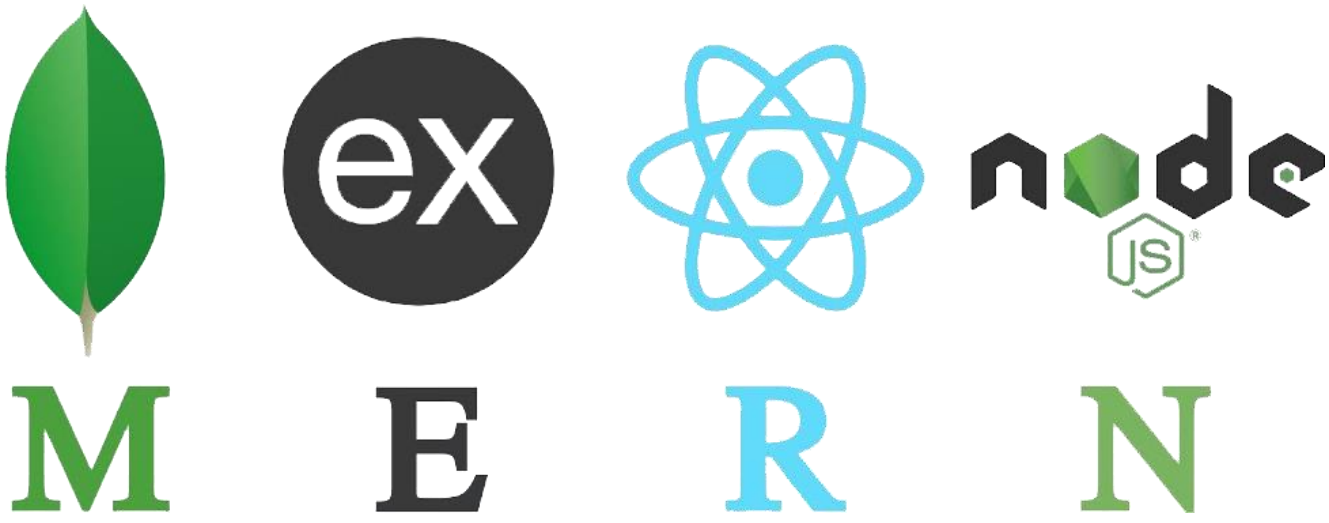


Fig.5 MERN

S

Note: Based on your requirements, you can install modules(by typing in **npm install module_name –save**) which will show up in package.json file.

Getting to know MERN Stack components:

1. MongoDB: Cross-platform Document-Oriented Database

MongoDB is a NoSQL database where each record is a document comprising of key-value pairs that are similar to JSON (JavaScript Object Notation) objects. MongoDB is flexible and allows its users to create schema, databases, tables, etc. Documents that are identifiable by a primary key make up the basic unit of MongoDB. Once MongoDB is installed, users can make use of Mongo shell as well. Mongo shell provides a JavaScript interface through which the users can interact and carry out operations (eg: querying, updating records, deleting records).

Why use MongoDB?

- Fast – Being a document-oriented database, easy to index documents. Therefore a faster response.
- Scalability – Large data can be handled by dividing it into several machines.
- Use of JavaScript – MongoDB uses JavaScript which is the biggest advantage.
- Schema Less – Any type of data in a separate document.
- Data stored in the form of JSON –
  1. Objects, Object Members, Arrays, Values, and Strings
  2. JSON syntax is very easy to use.
  3. JSON has a wide range of browser compatibility.
  4. Sharing Data: Data of any size and type(video, audio) can be shared easily.
- Simple Environment Setup – Its really simple to set up MongoDB.
- Flexible Document Model – MongoDB supports document-model(tables, schemas, columns & SQL) which is faster and easier.

Fig.6 mongo

2. **Express: Back-End Framework:**
   Express is a Node.js framework. Rather than writing the code using Node.js and creating loads of Node modules, Express makes it simpler and easier to write the back-end code. Express helps in designing great web applications and APIs. Express supports many middlewares which makes the code shorter and easier to write.
   Express.js, or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

Why use Express?

- Asynchronous and Single-threaded.
- Efficient, fast & scalable
- Has the biggest community for Node.js
- Express promotes code reusability with its built-in router.
- Robust API
- Create a new folder to start your express project and type below command in the command prompt to initialize a package.json file. Accept the default settings and continue.

Fig.7 express

3. **React: Front-End Library**
   React is a JavaScript library that is used for building user interfaces. React is used for the development of single-page applications and mobile applications because of its ability to handle rapidly changing data. React allows users to code in JavaScript and create UI components.

Why use React?

- Virtual DOM – A virtual DOM object is a representation of a DOM object. Virtual DOM is actually a copy of the original DOM. Any modification in the web application causes the entire UI to re-render the virtual DOM. Then the difference between the original DOM and this virtual DOM is compared and the changes are made accordingly to the original DOM.
- JSX – Stands for JavaScript XML. It is an HTML/XML JavaScript Extension which is used in React. Makes it easier and simpler to write React components.
- Components – ReactJS supports Components. Components are the building blocks of UI wherein each component has a logic and contributes to the overall UI. These components also promote code reusability and make the overall web application easier to understand.
- High Performance – Features like Virtual DOM, JSX and Components makes it much faster than the rest of the frameworks out there.
- Developing Android/Ios Apps – With React Native you can easily code Android-based or IOS-Based apps with just the knowledge of JavaScript and ReactJS.
- You can start your react application by first installing “create-react-app” using npm or yarn.

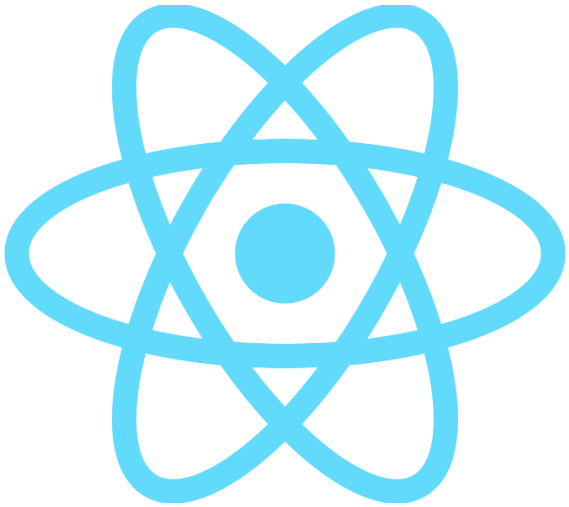


Fig.8 react

4. **Node.js: JS Runtime Environment**
   Node.js provides a JavaScript Environment which allows the user to run their code on the server (outside the browser). Node pack manager i.e. npm allows the user to choose from thousands of free packages (node modules) to download.

Why use Node.JS?

- Open-source JavaScript Runtime Environment
- Single threading – Follows a single-threaded model.
- Data Streaming
- Fast – Built on Google Chrome's JavaScript Engine, Node.js has a fast code execution.
- Highly Scalable
- Initialize a Node.js application by typing running the below command in the command window. Accept the standard settings.

Fig.10 nodejs

SWOT Analysis

SWOT analysis is a strategic planning and strategic management technique used to help a person or organization identify strengths, weaknesses, opportunities, and threats related to business competition or project planning. It is sometimes called situational assessment or situational analysis.

Strengths, Weaknesses, Opportunities, and Threats
SWOT stands for **Strengths, Weaknesses, Opportunities, and Threats**, and so a SWOT analysis is a technique for assessing these four aspects of your business.

How do you do a SWOT analysis?
Conducting a SWOT analysis

1. Decide on the objective of your SWOT analysis. ...
2. Research your business, industry and market. ...
3. List your business's strengths. ...
4. List your business's weaknesses. ...
5. List potential opportunities for your business. ...
6. List potential threats to your business. ...
7. Establish priorities from the SWOT.

Strengths:

- Merging users with similar interests.
- Quick login through Facebook and SMS.
- Working with non-relational database i.e MONGODB which is fast and reliable.
- Working with newest tech stack like MERN which has a very good industrial scope.
- Implemented industry best practice while coding.
- Version controlling through GitHub in our local environment.
- Worked with react which is a powerful frontend framework widely used.
- Created different database for different users which helps us to remove any proxy contents
- Stored passwords in encrypted form.

Weakness:-

- Group similar users will affect in our website traffic
- Categories can only be created by the admin which enables the user to post only specific categories
- Complexity and folder structure gets complicated.
- Dependency may become old after some time, so we may need to update the files after some time.
- Mondo doesn't work good when there's relational data.
- Need to depend on third party apps for some functionality of our project.
- Website may take to load sometimes.

Opportunities:

- Get More Clients to Your Existing Business
- Get Immediate Feedback
- Become a Better Writer
- Become a Published Author
- Become a Published Author
- Launch a Blog And Be An Influencer (Create a better world)

Threats:

- Injection Attack
- Broken authentication
- Cross site scripting
- Insecure direct object reference
- Security misconfiguration.

Algorithm used for Password storage:

Bcrypt: Bcrypt operates in a very similar manner to more traditional schemes based on algorithms like PBKDF2. The main difference is its use of a derived key to encrypt known plain text; other schemes (reasonably) assume the key derivation function is irreversible, and store the derived key directly.

The takeaway is this: bcrypt is a secure algorithm but remember that it caps passwords at 72 bytes. You can either check if the passwords are the proper size, or opt to switch to argon2, where you'll have to set a password size limit.

What is bcrypt ? bcrypt was designed by Niels Provos and David Mazières based on the Blowfish cipher>): b for Blowfish and crypt for the name of the hashing function used by the UNIX password system. crypt is a great example of failure to adapt to technology changes


```
_id: ObjectId("6255b99184a20251dc4baf5d")
avatar: "https://res.cloudinary.com/devatchannel/image/upload/v1602752402/avata..."
role: "admin"
type: "sms"
name: "+919138420233"
account: "+919138420233"
password: "$2b$12$7FqDrPzR6ObTWOsgfnyWle/xHjjulqE2lO/nLDqkQzsspk5ItIvHe"
rf_token: ""
createdAt: 2022-04-12T17:40:33.230+00:00
updatedAt: 2022-05-05T11:21:06.129+00:00
__v: 0
```


Fig.11 user data

UI (USER INTERFACE)

Technologies Used:

- HTML
- CSS
- Bootstrap
- Flask

Logo created using:

- Adobe Photoshop
- Adobe Illustrator

Website UI (Screenshots)

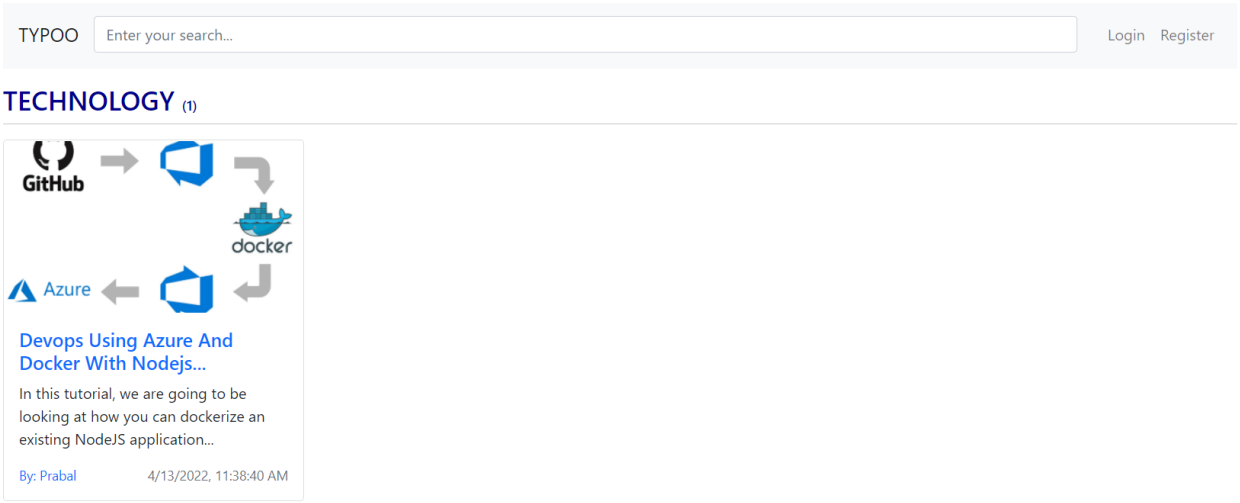


Fig.12 UI Homepage

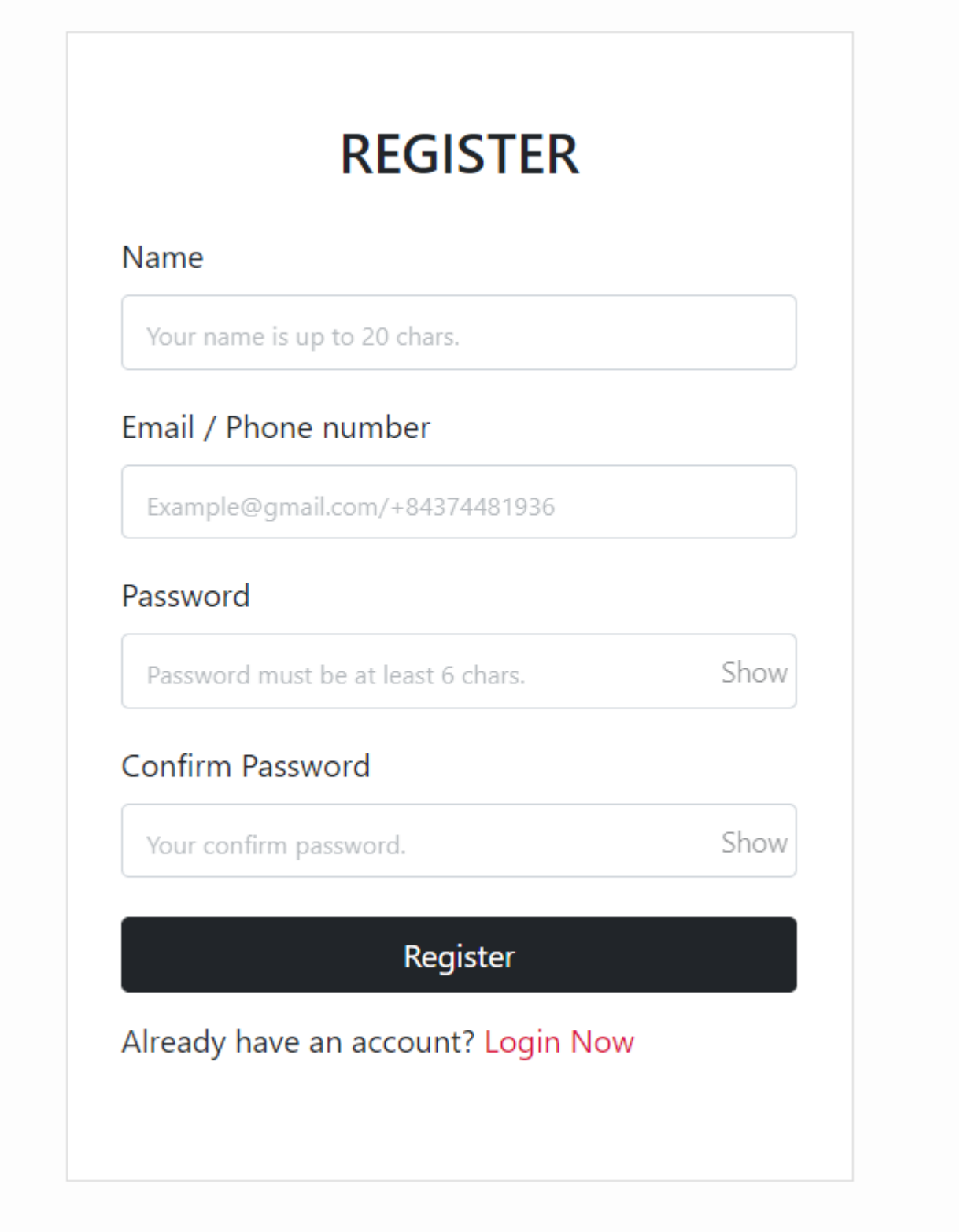


Fig. 13 Register Page

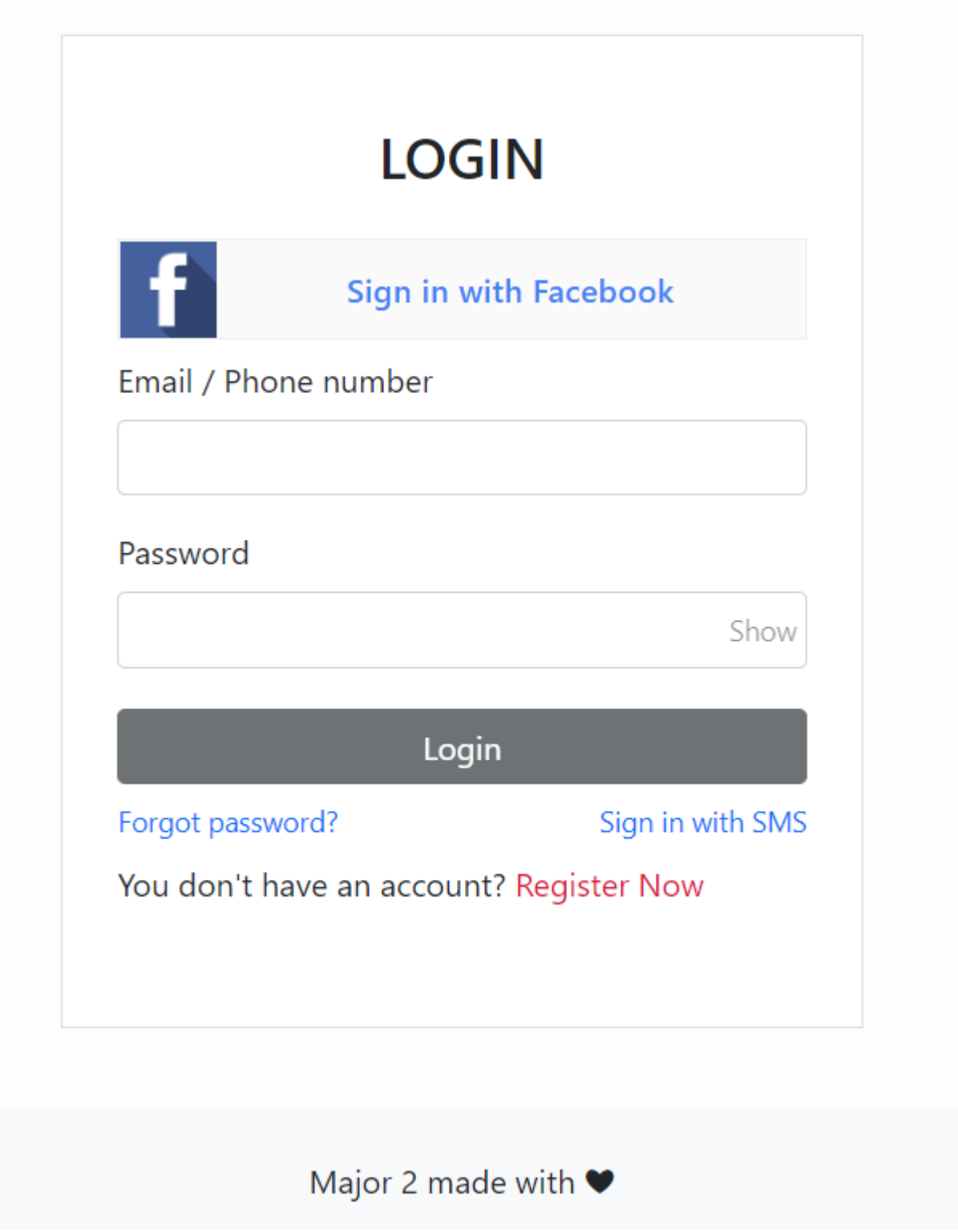


Fig.14 UI Login Page

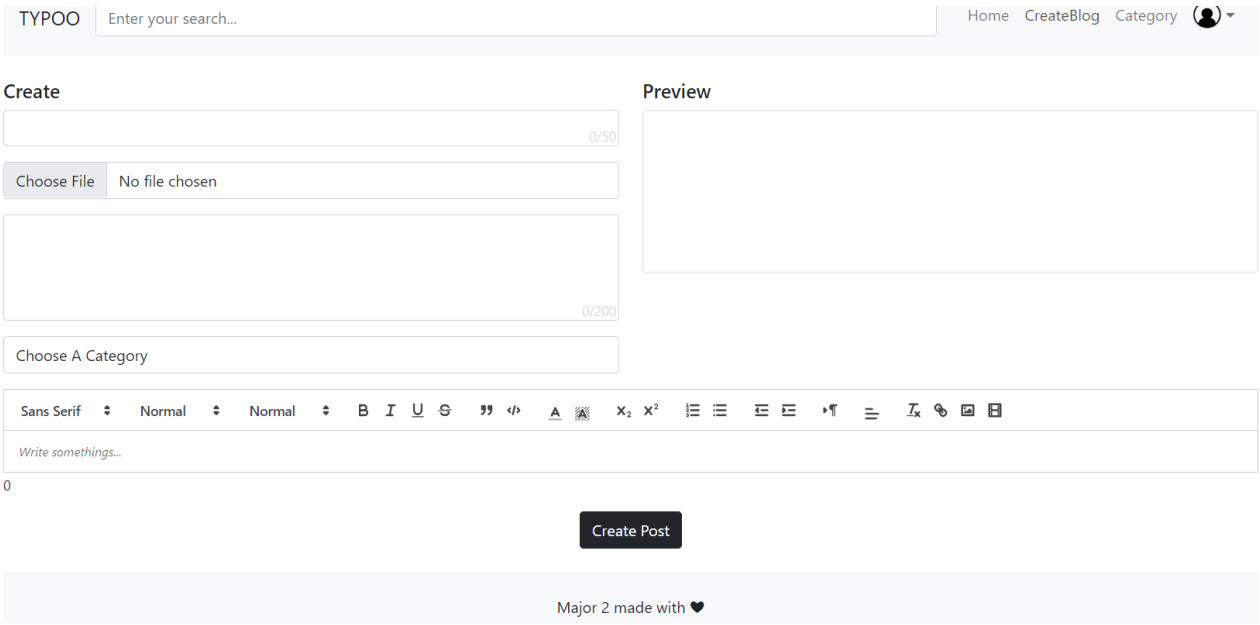


Fig.15 UI Create Post Page

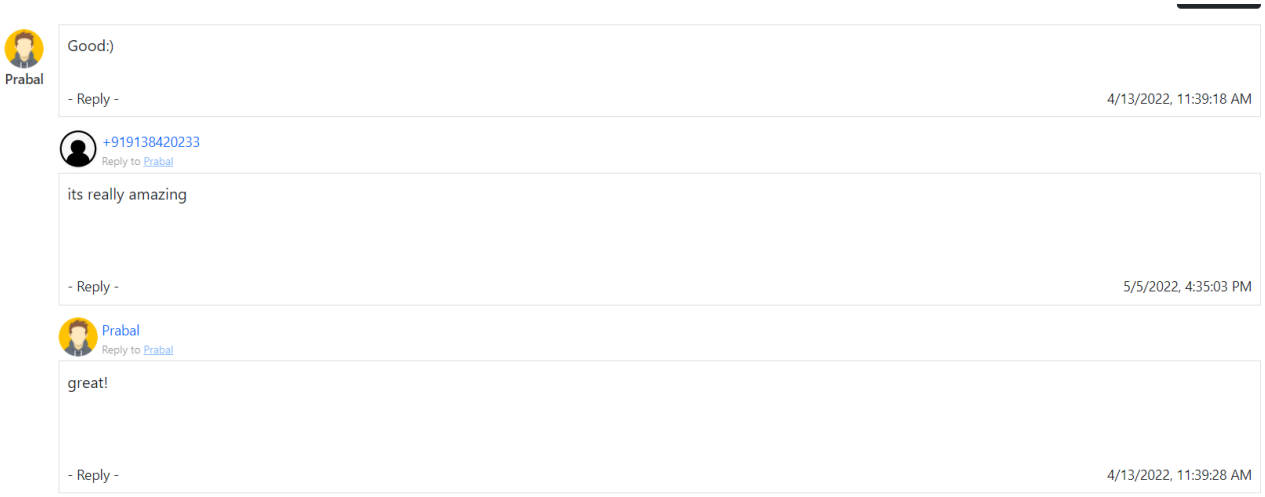
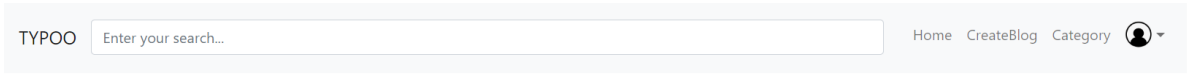


Fig.16 UI Posting Comments



Devops Using Azure And Docker With Nodejs

By: Prabal 4/13/2022, 11:38:40 AM

he application will listen on a port `3000` for any incoming requests and will map those requests against the corresponding route.

Docker should be installed in your system, if not refer to these commands, ignore them if already installed

```
1)sudo apt update
2)sudo apt install apt-transport-https ca-certificates curl software-properties-common
3)curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
4)sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
5)sudo apt update
6)apt-cache policy docker-ce
7)sudo apt install docker-ce
8)sudo systemctl status docker
```

Step 1: Project Setup

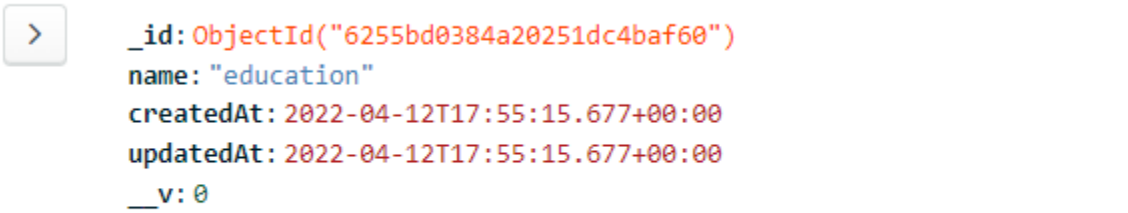
You can clone my repository or you can make any other nodejs application with a basic setup.If you are familiar with Node.js development, you'll know that the standard practice for this is to declare that your application requires `express.js` within the `package.json` in your application's directory.

Fig.17 UI Sample Blog

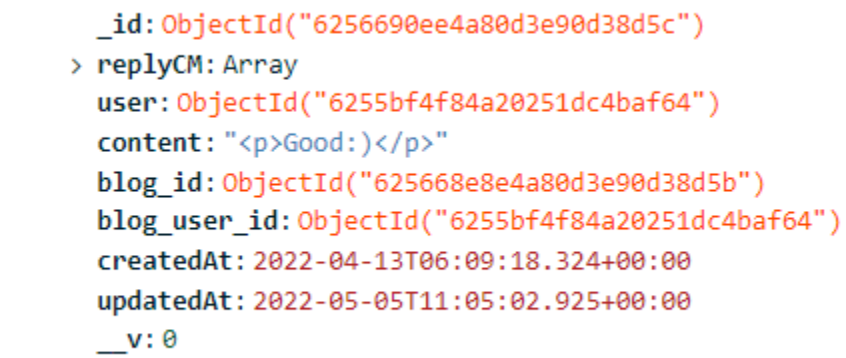
MongoDB database

User Data-Model

Categories Data-Model



Comments Data-Model



Users Data-Model


```
_id: ObjectId("6255b99184a20251dc4baf5d")
avatar: "https://res.cloudinary.com/devatchannel/image/upload/v1602752402/avata..."
role: "admin"
type: "sms"
name: "+919138420233"
account: "+919138420233"
password: "$2b$12$7FqDrPzR6ObTWOsgfnyWle/xHjjulqE2lO/nLDqkQzsspk5ItIvHe"
rf_token: ""
createdAt: 2022-04-12T17:40:33.230+00:00
updatedAt: 2022-05-05T11:21:06.129+00:00
__v: 0
```

CONCLUSION

As we know that most of the platforms are build up of combination of multiple component relying on different middleware that are API's.

Due to these API's a lot of work get easier as a lot of complex task are handled by these middleware.
People are able to develop product in short period of time with the help of them and are able to deliver the product more quickly as compared to when they have to create the product from scratch comprising all the features as well as all the different component of a product.

Keeping that in mind and further integrating with components like Node.js which is capable of server side scripting and MongoDB capable of fetching multiple request at the same time and giving a lot of flexibility with the Database to play around , combined these we can create applications that are more reliable and useful at the same time and can be scaled at any moment of time.

FUTURE SCOPE

We will enhance the functionality of all these operations we implemented in this project like -- Register, login with Email or Phone number. Quick login with Google, Facebook, SMS. Forgot password, reset password and register a new account by Email or SMS verification. Update personal information (name, password and avatar).Create new blog with React quill. Comment real-time with Socket.io. Pagination, search with autocomplete Mongodb.

Keeping that in mind and further integrating with components like Node.js which is capable of server side scripting and MongoDB capable of fetching multiple request at the same time and giving a lot of flexibility with the Database to play around , combined these we can create applications that are more reliable and useful at the same time and can be scaled at any moment of time.

PERT CHART

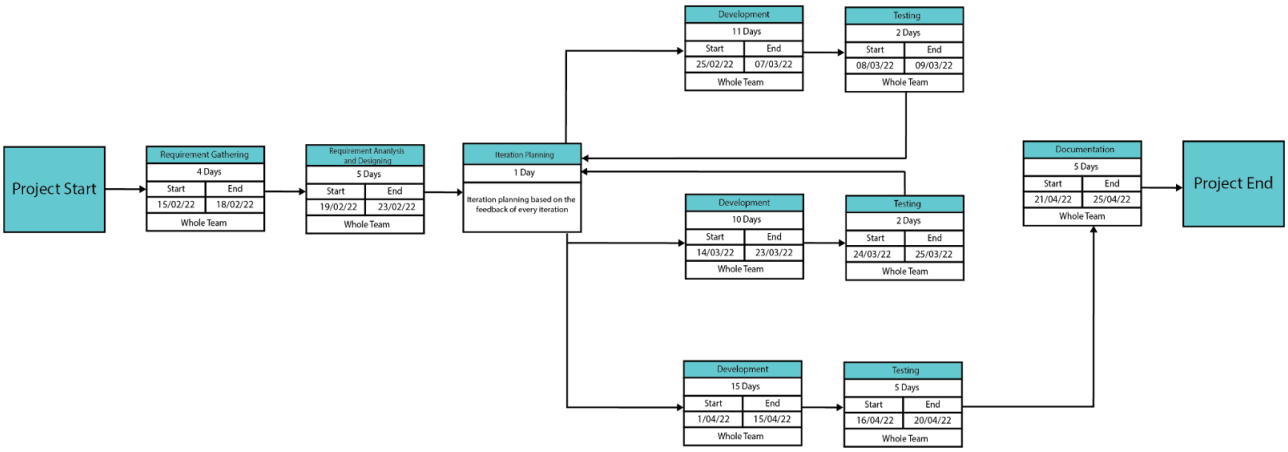


Fig.21 Pert Chart

REFERENCES

[1] Office for National Statistics, Internet users in the UK: 2016. Retrieved September 26, 2017, from https:// www.ons.gov.uk/ businessindustryandtrade/itandinternetindustry/bulletins/internetus ers/ 2016.

[2] Liang, L., Zhu, L., Shang, W., Feng, D., Xiao, Z. (2017). Express supervision system based on NodeJS and MongoDB. International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 IJERTV10IS060239 (This work is licensed under a Creative Commons Attribution 4.0 International License.) Published by : www.ijert.org Vol. 10 Issue 06, June-2021 714

[3] M. R. Solanki, A. Dongaonkar, A Journey of human comfort: web1.0 to web 4.0, International Journal of Research and Scientific Innovation (IJRSI), Volume III, Issue IX, pp. 75-78, 2016

[4] Javeed, A. (2019). Performance Optimization Techniques for ReactJS. 2019

[5] J. M. Spool, Content and design are inseparable work partners, 2014. Retrieved September 29, 2017, fromhttps://articles.uie.com/ content and design

[6] Bozikovic, H., Stula, M. (2018). Web design Past, present and future. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).

[7] Carter, B. (2014). HTML Architecture, a Novel Development System (HANDS): An Approach for Web Development. 2014

[8] Sterling, A. (2019). NodeJS and Angular Tools for JSON-LD. 2019 IEEE 13th

[9] Laksono, D. (2018). Testing Spatial Data Deliverance in SQL and NoSQL Database Using NodeJS Fullstack Web App. 2018

[10] Patil, M. M., Hanni, A., Tejeshwar, C. H., Patil, P. (2017). A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing Sharding in MongoDB and its advantages

[11] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, Image analogies, in Proc. 28th Annu. Conf. Comput. Graph. Interact. Tech., 2001, pp. 327-340.

[12] ”Tags used in HTML”. World Wide Web Consortium. November 3, 1992. Retrieved November 16, 2008.

[13] R. Irony, D. Cohen-Or, and D. Lischinski, Colorization by example, in Proc. Eurograph. Symp. Rendering, vol. 2. 2005, pp. 201-210.

[14] First mention of HTML Tags on the www-talk mailing list”. World Wide Web Consortium. October 29, 1991. Retrieved April 8, 2007.

[15] JavaScript specification. Retrieved from http://www.w3.org/standards/ webdesign/script, November 1, 2014. International Journal

[16] Bass, B.M. (1990), “From transactional to transformational leadership: learning to share the vision”, Organizational Dynamics, Vol. 18 No. 3, pp. 19-31.

[17] Bin, B.S. and Park, J.K. (2002), “An empirical study on the success factors of a small business startingup”, The Asia Pacific journal of Small Business, Vol. 24 No. 3, pp. 135-158

[18] Nodejs.org. 2020. Nodejs. Accessed on 28 April 2020 [online] Available at: https://nodejs.org/

[19] Jyoti Shetty, Deepika Dash, Akshaya Kumar Joish, Guruprasad C “Review Paper on Web Frameworks, Databases and Web Stacks” 2020 IRJET-V7141078

[20] Pro MERN Stack Full Stack Web App Development with Mongo, Express, React, and Node by Vasan Subramaniam

[21] Getting MEAN with Mongo, Express, Angular, and Node Book by Simon Holmes

[22] Angular.io,’AngularJS Documentation’ [Online]. Available: https://angular.io. [Accessed: May- 2021]

[23] MongoDB.com,’MongoDB Official’ [Online]. Available: https://www.mongodb.com/. [Accessed: May- 2021]

[24] Expressjs.com,’ExpressJS Official’ [Online]. Available: http://expressjs.com. [Accessed: May- 2021]

[25] ReactJS.org,’ReactJS official’ [Online]. Available: http://reactjs.org. [Accessed: May- 2021]

[26] Nodejs.com,'Nodejs Official' [Online]. Available: https://nodejs.org. [Accessed: May- 2021]

[27] K. Arnold. Programmers are people, too. Queue, 3 (5): 54-59, 2005.

[28] I. Balaban, F. Tip, and R. Fuhrer. Refactoring support for class library migration. SIGPLAN Not., 40 (10): 265-279, 2005.

[29] T. Biggerstaff. The library scaling problem and the limits of concrete component reuse. Software Reuse: Advances in Software Reusability, 1994. Proceedings., Third International Conference on, pages 102-109, 1-4 Nov 1994.

[30] J. Bloch. How to design a good API and why it matters. In OOPSLA '06: Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications, pages 506-507, New York, NY, USA, 2006. ACM.

[31] C. Bore and S. Bore. Profiling software API usability for consumer electronics. Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. International Conference on, pages 155-156, 8-12 Jan. 2005.

[32] S. Clarke and C. Becker. Using the cognitive dimensions framework to evaluate the usability of a class library. In Joint Conf. EASE & PPIG, Petre & D. Budgen (Eds), pages 359-366, 2003.

[33] D. Conway. Ten essential development practices. 14 July 2004.

[34] S. Demeyer, T. D. Meijler, O. Nierstrasz, and P. Steyaert. Design guidelines for tailorable frameworks. Commun. ACM, 40 (10): 60-64, 1997.

[35] B. Ellis, J. Stylos, and B. Myers. The factory pattern in API design: A usability evaluation. In ICSE '07: Proceedings of the 29th International Conference on Software Engineering, pages 302-312, Washington, DC, USA, 2007. IEEE Computer Society.
[36] M. Fayad and D. C. Schmidt. Object-oriented application frameworks. Commun. ACM, 40 (10): 32-38, 1997.

[37] D. Garlan, R. Allen, and J. Ockerbloom. Architectural mismatch or why it's hard to build systems out of existing parts. In ICSE '95: Proceedings of the 17th international conference on Software engineering, pages 179-185, New York, NY, USA, 1995. ACM.

[38] E. R. Harold. XOM design principles. Extreme Markup Languages, 2-6 August 2004.

[39] J. Henkel and A. Diwan. CatchUp!: capturing and replaying refactorings to support API evolution. In ICSE '05: Proceedings of the 27th international conference on Software engineering, pages 274-283, New York, NY, USA, 2005. ACM.

[40] M. Henning. API design matters. Queue, 5 (4): 24-36, 2007.

[41] S. Henninger. Retrieving software objects in an examplebased programming environment. In SIGIR'91: Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval, pages 251-260, New York, NY, USA, 1991. ACM.

[42] R. Holmes, R. Walker, and G. Murphy. Approximate structural context matching: An approach to recommend relevant examples. Software Engineering, IEEE Transactions on, 32 (12): 952-970, Dec. 2006.

[43] D. Mandelin, L. Xu, R. Bodík, and D. Kimelman. Jungloid mining: helping to navigate the API jungle. In PLDI '05: Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation, pages 48{61, New York, NY, USA, 2005. ACM.

[44] S. McLellan, A. Roesler, J. Tempest, and C. Spinuzzi. Building more usable APIs. Software, IEEE, 15 (3): 78- 86, May/Jun 1998.

[45] Microsoft Developer Network. Designing .NET class libraries: Designing progressive APIs. MSDN Online Chat, 02 March 2005.

[46] L. R. Neal. A system for example-based programming. In CHI '89: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 63-68, New York, NY, USA, 1989. ACM.

[47] L. R. Neal. A system for example-based programming. In CHI '89: Proceedings of the SIGCHI conference on Human factors in computing systems, pages 63-68, New York, NY, USA, 1989. ACM.

[48] D. L. Parnas. On the criteria to be used in decomposing systems into modules. Commun. ACM, 15 (12): 1053- 1058, 1972.

[49] D. L. Parnas and D. P. Siewiorek. Use of the concept of transparency in the design of hierarchically structured systems. Commun. ACM, 18 (7): 401-408, 1975.

[50] B. Pryor. Simple concurrency guidelines. Ben Pryor's Blog, March 2008.

[51] Y. Smaragdakis and D. Batory. Mixin layers: an objectoriented implementation technique for refinements and collaboration-based designs. ACM Trans. Softw. Eng. Methodol., 11 (2): 215-255, 2002

[52] J. Spolsky. How Microsoft lost the API war. In Business of Software 2008, a JOEL ON SOFTWARE Conference, Boston, MA, United States, 13 June 2004.

[53] J. Stylos and S. Clarke. Usability implications of requiring parameters in objects' constructors. In ICSE'07: Proceedings of the 29th International Conference on Software Engineering, pages 529-539, Washington, DC, USA, 2007. IEEE Computer Society.

[54] J. Stylos and B. Myers. Mapping the space of API design decisions. Visual Languages and Human-Centric Computing, 2007. VL/HCC 2007. IEEE Symposium on, pages 50-60, 23-27 Sept. 2007.

[55] J. van Gurp and J. Bosch. Design, implementation and evolution of object oriented frameworks: concepts and guidelines. Softw. Pract. Exper., 31 (3): 277-300, 2001.

[56] T. Xie and J. Pei. MAPO: mining API usages from open source repositories. In MSR '06: Proceedings of the 2006 international workshop on Mining software repositories, pages 54-57, New York, NY, USA, 2006. ACM.

[57] Z. Xing and E. Stroulia. API-evolution support with DiffCatchUp. Software Engineering, IEEE Transactions on, 33 (12): 818-836, Dec. 2007.